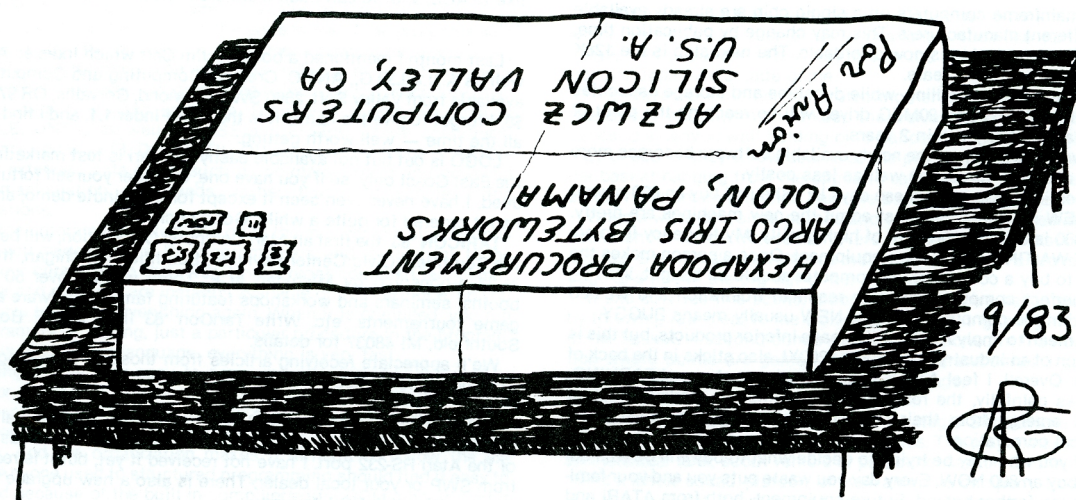


ACE
ATARI
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

OCTOBER, 1983

Mike Dunn & Jim Bumpas, Editors



ACE THIRD ANNIVERSARY!

Happy birthday to us and to those who have been with us from the start. This issue marks three years of publication for the ACE Newsletter, which began with the October, 1980 issue. I've only been with ACE since July, 1981 myself.

In the past three years, users have pioneered uses of the Atari computer for which Atari, Inc. and the rest of us never imagined possible. Many of those users continue to share their discoveries in this newsletter.

Now there are a half-dozen different Atari computers and thousands of software titles, and peripherals are multiplying explosively. Hobbyist and user explorations of the limits of these items will range even more widely in the future. We will continue sharing these pioneer efforts in the pages of the ACE Newsletter.

NIBBLING BYTES

by Kirt E. Stockwell

An old Arab curse goes like this: "May you live in interesting times." Apparently they preferred not having to keep up on current events. Since we DO live in interesting times, we might as well make the best of them.

You are probably aware this has been an extremely interesting year. ATARI has announced a new line of computers. ATARI has made many personnel changes, and ATARI stock has done interesting things. The industry as a whole is buzzing with activity, not the least of which is the effort to develop a computer better than the one the competition is going to release soon. The micro computer makers are all trying to beat the pants off of each other. Business computer makers are at each others' throats as usual. The entire nation of JAPAN has challenged the US to a race to see who can develop the next generation of SUPER-COMPUTER.

You can read about all of this in any weekly magazine, but have you thought about how it will affect you? Consider some of the following items:

32 bit mainframe computers on a single chip are already available from 3 different manufacturers. This may change by publication time.

The 64K memory chip is now affordable. The next step is the 128K chip, expected within 2 years.

Disk drive prices are falling, while densities and storage capacities are growing. 10MEG and 20MEG drives will be reaching the point of common affordability within 2 years.

With every passing year the newly available software becomes more powerful and easy to use, as well as less costly.

The question I hear most these days is: "Should I buy an 800 or wait for the NEW machines?" I must admit the new machines are pretty. But the 800 is a proven product of high quality. My tendency is to advise NOT WAITING for any new equipment. Once a person makes the decision to buy a computer, the computer should be bought. Waiting for the perfect computer will only result in frustration and wasted time. Within the computer industry, NEW usually means BUGGY.

I don't mean to imply ATARI will release inferior products, but this is a reflection of an industry reality. The 1200XL also sticks in the back of my mind. Overall, I feel ATARI has spent the past year LEARNING, sometimes painfully, the realities of the computer market. I expect they will emerge from their problems as a leaner, cleaner, more methodical corporation.

Any of you who may be trying to decide what to buy will be well advised to buy an 800 NOW. Every day you waste puts you and your family that much farther behind. Future equipment, both from ATARI and from the various mainframe companies, promises to be quite advanced. What you have now, or buy in the near future, will not be particularly important. What will be important is whether or not you understand the equipment and the market.

News and Reviews

by Mike Dunn, Co-Editor

Atari computers are being sold now for amazing low prices. I've heard of \$180 for an 800 with BASIC and 48K, and a 1200 for \$219. Speaking of the ill-fated 1200XL, Atari has made available for 1200 owners who are having trouble running third party software a pre-boot disk turning the 1200 OS to a 800 OS. Contact customer service for details.

Have you read the Oct issue of Consumer Reports? They have been doing a several part article on computers and peripherals; the first, last month was not particularly favorable to Atari. This month, they have upgraded the Atari considerably because of **AtariWriter**, which they consider one of the best and easiest to use wordprocessors available — they even rate it in the league of WordStar, and other big league programs. They think AtariWriter is worth buying an Atari for alone! APX now has the AtariWriter **printer drivers** available for only \$15 and they are much better than I expected them to be. New printers are added all the time; they now include the Epson MX80, FX8, RX80, Gemini10, IDS480, MX100, NEC8032a and the C. Itoh Prowriter. Each driver allows the use of all the special features of the printer; they even allow right justification of proportional print. None of the other wordprocessors for the Atari have this option; and until they do, it does make the **AtariWriter** a best buy if this is important to you. Since these files are automatically booted in, it also makes this wordprocessor the easiest to use for the printers noted above.

The most impressive new program I've seen in the entertainment area ever is the fantastic new program by Chris Jochumson and Doug Carlston of Broderbund called **The Arcade Machine**, (17 Paul Drive, San Rafael, CA 94903, \$60. Although it arrived too late for a full review — see next issue — this incredible new "language" allows you to make arcade quality games of your own design with great ease. It includes 4 made games on files as examples; the manual tells you how to modify them. You can easily create shapes, backgrounds and title pages, then make the shapes move in any path you wish. You can add all kinds of "canned" sound effects as well as make your own, including music. You can make up to 5 levels for each game, have various options, etc. The games that are included are similar to commercial quality games. You can easily test and change them to your hearts content; then, when all is ready, automatically create a "boot disk" and send them to your friends (or ACE!) to amaze your friends. It is very easy to create a game like, say, "Joust". If you create something you really like and think is great, you can send it to Broderbund and win prizes or even a contract as a game creator. If you like **Pinball Construction Set** by Electronic Arts, you'll go wild over this one! A "must buy" for Arcader's — please send the ones you really like to ACE, and we can trade them around!! More next month on this one.

Last month I mentioned a book by Jim Carr which indexes ACE, as well as A.N.A.L.O.G., ANTIC, Creative Computing and Compute!, and available from Valley Soft, 2660 SW DeArmond, Corvallis, OR 97333 for \$6. I forgot to give the name: it is the **Soft Finder 1.1**, and I find I use it all the time — well worth getting.

LOGO is out but not available easily — Atari is test marketing it on the East Coast only; so if you have one, consider yourself fortunate indeed. I have never even seen it except for a 5 minute demo, and have had it ordered for quite a while. Good Luck.

TARICON '83, the first all Atari Computer Convention, will be held at the Southfield Civic Center Pavillion, Southfield, Michigan, from Oct 21-23. Sponsored by Michigan ACE. There will be over 50 exhibit booths, seminars and workshops featuring famous software authors, game tournaments, etc. Write TariCon '83 Info, MACE Box 2785, Southfield, MI 48037 for details.

We'll appreciate receiving articles from those of you going to this exciting conference!!

The **ATR 8000** company, SWP, has finally released the final version of MYDOS which allows use of double-sided drives, as well as the use of the Atari RS-232 port. I have not received it yet, but it is ready. \$30 from SWP or your local dealer. There is also a new upgrade EPROM (3.2) to allow this DOS to work.

Last month, I mentioned my **Austin Franklin 80 Col Board** (Austin Franklin Assoc., 43 Grove St., Ayer, MA 01432). After talking to Mr. Franklin, I found out the problem was my 32K memory board. Since I had one of the first 32K boards made, it used too much power, and made the image unstable. He is going to send me his 48K board to show me the difference. I have not received it yet, but will report on it when I do.

The cover cartoon is by a new member who is stationed somewhere in the tropics, A. Grimm Richardson. Mr. Richardson is a prolific writer for various publications, but, unfortunately, all the ones he has written for have gone out of business. His cartoon is somewhat obscure, but very funny when you figure it out. The optional caption is "And where do you think computer hardware and software gets them?" Remember, the artist is from the tropics, where they really grow well.

Ultra Disassembler

Adventure International, Box 3435, Longwood, FL 32750, \$50

Written by Ralph Jones, this new disassembler is one of a series of utilities, including Diskey, called the Ultra series by AI. If the future one are as good, Atari owners can rejoice.

Disassemblers all translate machine language (binary load) files into assembly language so you can see "List" the program. The Ultra-disassembler adds some amazing features. First, it allows you to take the assembly language listing and put it into the assembler of your choice for modification and re-assembling it to machine code.

Although specifically designed to use the Atari MacroAssembler, there is a utility program allowing you to use almost any of the other available assemblers. The assembly language listing produced is formatted in pseudo-code with standard Atari system labels. As an example, the memory location \$02E5 is called MEMTOP by Atari, since it is the memory location of the Atari operating system which gives the location of the top of memory of the system. When a program is disassembled, instead of listing something like STA \$02E5, which is difficult to understand, it will say STA MEMTOP. This allows for much easier understanding of the listing. A great tool for learning machine language by example.

ACTION!

O.S.S., 10379 Landsdale AVE., Cuperino, CA 95014, \$99 ROM

From the originators of Atari BASIC, as well as many other languages for the Atari: BASIC A+, C, MAC-65, etc., comes a new, original language available only for the Atari. ACTION! is in the new Precision Software Tools series which are on the OSS "SuperCartridge", allowing 24K on one ROM by bankswitching. This cartridge includes the language ACTION!, a Text Editor, a Compiler, monitor and a library of useful procedures all on one cartridge! Switching back and forth is instantaneous.

The language is a extremely high speed, high level, structured language combining the best features of C, Pascal, Ada and BASIC. It runs 100 to 200 times faster than BASIC. Although it is compiled, the compiler is built in and does not have to be loaded in.

You begin programming by using the powerful editor. This built-in screen editor uses the horizontal and vertical scrolling window technique allowing a line to be 120 characters wide or more — I recommend you set the line width to the width of your printer. The editor has many features of a good text editor, such as Search and Replace, ability to move text blocks, 2 windows, various types of deletions, restoring deletions if you change your mind, scrolling up and down, etc. The ACTION! language does not require line numbers.

ACTION! is a structured language similar to C. It is actually easy to program in — the manual includes a section on translating BASIC to ACTION!, and it was very easy to translate some of the programs written in C in Byte. The cartridge includes many procedures in its library to simplify programming. The language includes many bit manipulation operators. You begin programming by naming the procedure, then declaring data types and variables. Structured statements include IF...WHILE...UNTIL, as well as ELSE and ELSEIF. These conditional statements end with FI. DO...UNTIL...OD are used for loops. Available also are extended data types such as pointers, arrays and powerful record manipulation commands, rather than just simple string manipulations.

After the program is written, you go to the monitor and start to compile it. If you made an error, the program stops, and you go back to the editor; your cursor is on the error. Unlike most compiled languages, the editor and compiler are instantly available on the cartridge, so you do not have to re-load everything again. When it is compiled, you run the program, or, if testing, just a particular procedure in the program. Since it is a compiled language, everything is very fast.

The language is very easy to program with, probably easier than BASIC for a beginner. The manual, however, makes it very difficult for a beginner to learn. There are too few examples, and several of them have typographical errors that are frustrating for a beginner. There is no index, and the organization makes it difficult to follow the examples. Anyone used to structured programming should find it easy to learn, and because of the built in compiler/editor ROM, should find it a pleasure compared to Pascal and C.

Compiled programs need ACTION! to run — there does not appear to be a run-time utility. O.S.S. will be releasing a disk of ACTION! programs which should go a long way toward helping us learn. This could be a very popular language, especially if it is made available for use with other micro-computers. If any of you develop some programs, especially comparing BASIC with ACTION!, please send them in and we can start a regular column.

— M. Dunn

BUMPAS REVIEWS

This month I have five games, all of which I rate a 10.

Excalibur, \$29.95 from APX, is the latest and greatest creation from the mind of Chris Crawford. You are Arthur with the quest to become King of England forever. But first, the other 15 kings must pledge fealty to you. One or two of them might even rise to compete with you during the game for the control of all England. You must use your prestige, the Knights of the Round table and your army of men at arms, and Merlin to complete the quest.

The game includes three modules: A beautiful map of all England which scrolls across more than a dozen screens; Camelot castle with five separate rooms full of objects to manipulate; and the field of battle where the knights and men at arms appear in battle array.

Each king has his own castle, and Arthur can only enter the castle of a vassal. But with Merlin's help, he can "see" into the castle of any king to watch how the computer manipulates treasury, taxes and army. Arthur keeps a strategic map in his Throne Room which highlights each kingdom in a distinct color indicating a vassal, neutral, tributary or hostile king.

This is a real-time strategic game. Weeks pass without regard to how fast or slow you do your moves. An "Intermission" is available in case you are called away from the game briefly. One game in progress may be saved on a disk. I'm interested in exchanging information on the experiences you will have in this game. I've already united England under Arthur once, but I'm sure I did not do it the most efficient way. I've never seen a strategic game I enjoy more than this one.

Dark Crystal is a graphics adventure game on 3 disks from Sierra OnLine for \$60. I'm not a real fan of adventure games, but you don't have to be one to enjoy this game. Fans of the movie will enjoy how closely this game parallels the movie. This game makes you the character Jen in the movie. As the last surviving Gelfling, your quest is to find and repair the broken shards of the Crystal. All the baddies from the film are here to stop you.

Operation Whirlwind from Broderbund for \$39.95, puts you in command of a German mechanized battalion with the task of driving 15 km through Russian lines to take and hold a town. The map scrolls across nearly 10 screens of finely detailed terrain. Enemy units are hidden until they move or fire. There is some real-time action during the combat phase. The longer you take to complete your combat, the longer you are bombarded by Russian artillery. You must also be concerned with the range and effect of your weapons against the various targets. A game in progress can be saved at the end of any turn. Player performance is evaluated by headquarters and reported each turn.

POOYAN, \$29.95 from Datasoft is a joystick game for small children which will give the rest of us a challenge, too. The name means "piglet" in Japanese. This may mean the game is an import of a popular Japanese arcade game.

The player is a mother pig protecting her piglets from wolves. She has arrows and raw meat to keep the wolves away. The wolves must descend a canyon wall holding onto balloons to get at the piglets. As they descend, they throw acorns at the pig, trying to knock her out of her basket hanging from the opposite canyon wall. Two piglets with a pulley mechanism attached haul the basket up and down.

Once a wave of wolves passes and one has pig lives remaining, the scene shifts to the wolves' lair. Here the wolves try to ascend to the cliff top. Seven wolves at the top will be able to push a stone over the edge and knock the pig out of her basket.

If she survives this scene, a bonus scene allows her to gain points by throwing raw meat to the wolves without risking being knocked out of her basket. A second bonus scene is mentioned in the documentation where points are gained by shooting strawberries. I guess you need more skill than I have to get to that page.

COMBAT LEADER for \$39.95 from Strategic Simulations, Inc. is a joystick game simulating small-unit tactics. The player may command an individual tank, scout vehicle or personnel carrier. Or one may command a whole squad, platoon, or even an entire company. Enemy units and any units not commanded by the player will be commanded by the Atari.

The map scrolls vertically across more than six screens, covering a 40x77 grid. The terrain shows woods, hills with topographical lines, rocky areas and depressions (these look like shell-holes to me). The map is predominantly green or olive-drab depending upon the scenario.

The little tanks show turning turrets; the carriers look like boxes; the infantry are only little "+"s. Shooters show little muzzle flashes of red; targets blow up in a bigger red fireball, leaving wrecks which look like ants or beetles. Each weapon produces its distinctive sound. Engine noises accompany movement.

Dozens of "canned" scenarios are available on the disk by combining various mixtures of terrain, weapons systems, and missions. Or one may design custom scenarios to one's own specifications by responding to a couple of dozen prompts setting the various terrain densities, unit sizes, weapons characteristics, pace of the game and a random number from 1 to 8 which alternates the positioning of the various terrain features.

A chart of dozens of tanks, scout vehicles and carriers is provided so one may create scenarios using historical vehicles with characteristics which conform to game parameters. It's been a long time since I've enjoyed playing a wargame more than this one.

—Jim Bumpas

FROM THE ACE LIBRARY

When Aaron and I took over the library duties from Chuck and Jody Ross, we thought it was going to be an easy job. Just make a few disks and mail out a few library lists every month and that's it. Boy, were we wrong! The ERACE group (Educational Researchers of ACE) snowed us under with five new disks and the editors of the newsletter added the Best of ACE Vol. #6.

The mail has been coming in from all over the world. Besides the USA and Canada, we have heard from South Africa, Sweden, Indonesia, Japan, New Zealand, Australia, and a research ship somewhere in the Atlantic Ocean. We really enjoy hearing from all of you. Please excuse us if we may be late in sending a reply.

Many of you have written asking for specific programs to do specific jobs. When asking for programs, please be sure to include what type of equipment and memory capacity you are using. This way we can do a better job helping you obtain the software you need. If you have a question we can't answer, we will pass it along to one of the many experts in the club or refer you to someone who may be able to help.

Now, to the newest disks ...

The Best of ACE Vol. #6 for \$15.00 is now available. Programs include most of the games and a few of the utilities which have been in the newsletter from June through September, including the latest version of TINYTEXT, RAMTEST, RUBIC'S CUBE, TITAN LANDER, and many others.

EDUCATION DISK #4 (by ERACE) for \$8.00 and includes MATHWARS, GEOQUIZ (a geography quiz), HANGQUOTE (a hangman type of game using famous quotations instead of single words), and ODDEVEN, PLURALS, NOUNS, and VERBS. There are 10 programs on this disk from elementary to advanced levels. A more complete description is available on the library list.

We also have added four disks made available to us from the San Mateo County Office of Education and Computer Using Educators. Called SOFTSWAP (\$8.00 each), there are 25 programs on these four disks. You will have to write for the library list for a complete description. All of these are very well done. I am sure you won't be disappointed.

The newest library list (revised 9-15-83) is off the press and ready to mail. To obtain your copy please send 25 cents (50 cents overseas) in stamps or coin to Ron and Aaron Ness, ACE Exchange Library, 374 Blackfoot Street, Eugene, OR 97404.

— Ron Ness

400 KEYBOARDS

I did purchase the S/Ware keyboard. Unfortunately, the amount of time required to complete all the wiring makes this a most cost ineffective product as far as I am concerned. The documentation is very well put together. The time, though, to solder every switch and run the various loops of wires will likely take several hours. Of additional note, the product does not come with a case for the keyboard, which will likely add an additional \$25 to the cost. Since the vendor did not have a telephone number to call for problems, and implied in the published limited warranty that returns are accepted for defects only, I was not sure what problems I was going to have in returning the product. Fortunately, the vendor was gracious enough to accept my return of his product minus a restocking fee.

Next I obtained the "Sidewriter" keyboard manufactured by Screensonics. This is an excellent keyboard which is connected to the Atari by several feet of flat cable, with pin connections to both the keyboard and the Atari. It is sufficiently heavy to offer an excellent "feel" when placed on one's lap. The board comes prewired and in a case. The only wiring required is to connect to the Atari keyboard pins, which should not take more than 1 hour. My only critique is the lack of a keypad option on the keyboard. The price is approximately \$170.

I also just purchased ATSPELLER by APX. This is a word dictionary utilizing Atari DOS.

—Elliot H. Fishkin, M.D.

Stereo Cassette Interface

Essence Peripheral Systems. Model 600 for \$50.00.

Here is a new product for all cassette users. This interface allows you to use your stereo cassette deck to load and save your programs. The stereo cassette deck gives you clearer, sharper and better all around copies of your programs.

This IF comes complete & ready to use except for a set of patch cords which can be bought at Radio Shack. EPS even gives you the RS part no.

Instructions are quite clear and once you hook up the patch cord to the interface, the cassette deck and then to the computer you are ready to go. The interface has only one switch to use. One position is CLOAD, the other CSAVE. The only thing left to do now is to load one of your programs in with the deck and turn everything on as you normally might, except the interface is in the CLOAD mode or load a blank tape and use the CSAVE mode.

One of the things I found annoying was that my cassette deck was with the Hi-Fi equipment in another part of the house. And to bring in a portable deck to use just means more equipment around the computer. For those of you who have Hi-Fi or cassette decks close by, this interface makes or plays cassettes better than the 410 Program Recorder does. It is a good investment for those who use a lot of cassettes and want to make sure they have the best possible recording or playing of their programs.

The ACE librarians will use this interface to make cassettes for the library.

—Larry Gold

MEETING!

7.30 pm, Wednesday, Oct. 12

FORUM #308, LCC

John Atack will demonstrate his Quasimoto.

Robert Browning: Upper & LowerCase

```

0 REM *****
1 REM ** ACE NEWSLETTER **
2 REM ** 3662 VINE MAPLE **
3 REM ** EUGENE, OR 97405**
4 REM ** $10 YR **
5 REM *****
6 REM *****
7 REM * GRAPHICS 1 & 2 TEXT *
* UPPER AND LOWERCASE *
8 REM * by ROBERT BROWNING *
9 REM * OCTOBER 1983 *
10 REM *****
11 REM ***** SETUP *****
20 DIM ANS$(12),ANS2$(12),P$(40),QUES$(1)
30 CLOSE #1:OPEN #1,4,0,"K:"
40 POKE 106,PEEK(106)-5:GRAPHICS 0
50 SL=(PEEK(106)+1)*256
60 POKE 752,1
70 POSITION 9,5: "Give me 22 seconds
to move and redefine t
h
e
character set."
80 POSITION 9,12: "When you want to e
nd the program, type 'end
,
for your name,"
85 POSITION 20,14: "or press the
[ESC] key."
90 FOR MM=0 TO 1023:POKE SL+MM,PEEK(57
344+MM):NEXT MM
100 FOR MM=256 TO 472:POKE SL+MM+256,P
EEK(57344+MM):NEXT MM
110 FOR R=1 TO 4:READ A:READ B:FOR MM=
1 TO 8:POKE SL+A+MM+256,PEEK(57344+MM+
B
):NEXT MM:NEXT R
120 FOR I=256 TO 263:POKE I+SL+256,0:N
EXT I
125 GOSUB 900
130 REM ***** YOUR PROGRAM GOES HERE *
*****
140 ? " %":POSITION 8,5
145 POKE 755,1:POKE 752,1
150 ? "Do you want to see each
character and it's ATASCII
number?"
160 POSITION 8,9: ? "(Y OR N)
)"
170 MAX=1:LTR=1:XX=24:Y=9:GOSUB 380:OU
ES$=ANS2$
175 GRAPHICS 2:POKE 752,1
176 POKE 756,SL/256+2
180 ? #6: " %": ? " %"
190 POKE 756,SL/256+2
200 POSITION 1,1: ? #6: "what is your na
me?"
210 POSITION 1,2: ? #6: "WHAT IS YOUR NA
ME?"
220 POSITION 1,3: ? #6: "WHAT IS YOUR
NAME?"
230 POSITION 1,4: ? #6: "WHAT IS YOUR
NAME?"
240 POSITION 1,6: ? #6: "What is your
name?"
250 POSITION 1,7: ? #6: "WHAT IS YOUR NA
ME?"
260 POSITION 1,8: ? #6: "WHAT IS YOUR
NAME?"
270 POSITION 1,9: ? #6: "WHAT IS YOUR
NAME?"
280 IF QUES$="Y" THEN FOR R=0 TO 255:P
OSITION 1,5: ? #6:CHR$(R):FOR W=1 TO 50
:
? R:NEXT W:NEXT R
290 FOR R=1 TO 500:NEXT R: ? #6: " %"
300 ? #6: "What is your name?"
310 ? " Use the shift key for capitals
."
320 MAX=10:XX=4:Y=5:LTR=2:GOSUB 380
330 REM THIS LINE GOES IN PLACE OF AN
INPUT STATEMENT:MAX=X:LTR OR NUM=N:XX=
N
:Y=N:GOSUB 180:POKE 764,255
340 REM GOSUB 180 FOR LETTER ROUTINE,
190 FOR NUMBER ROUTINE
350 ? ANS$,ANS2$:FOR W=1 TO 500:NEXT W
: ? " %": ? #6: " %":IF ANS2$(">")END" AND
A
NS2$(">")END" THEN 180
360 ? #6: " %":X=6:Y=5:P$=" \000 023
":GOSUB 750:GOTO 360
370 REM ***** GETKEY *****
380 NUM=0:GOSUB 400:RETURN
390 LTR=0:GOSUB 400:RETURN
400 BREAK=PEEK(16)-128:IF BREAK<0 THEN
GOSUB 430:RETURN
410 POKE 756,SL/256+2
420 POKE 16,BREAK:POKE 53774,BREAK
430 FOR R1=1 TO 500:POKE 702,0:POKE 69
4,0:POKE 764,255:SP=0:IF LTR=1 THEN PO
K
E 702,64
440 ANS$="":ANS2$=""
450 FOR R2=1 TO 500
460 GET #1,AV:KEY=PEEK(754):IF KEY=52
THEN 660
465 IF KEY=28 THEN 360
470 IF AV<32 THEN NEXT R2
480 IF AV>32 AND AV<39 THEN NEXT R2
490 IF AV>39 AND AV<46 THEN NEXT R2
495 IF AV>46 AND AV<48 THEN NEXT R2
500 IF AV>47 AND AV<58 AND LTR THEN NE
XT R2
510 IF AV>57 AND AV<65 THEN NEXT R2
520 IF AV>57 AND AV<155 AND NUM=1 THEN
NEXT R2
530 IF AV>90 AND AV<97 THEN NEXT R2
540 IF AV>122 AND AV<155 THEN NEXT R2
550 IF AV>155 THEN NEXT R2
560 GM=PEEK(87)
570 IF GM>0 AND AV<39 THEN AV=28
580 IF AV=155 AND SP=0 THEN NEXT R2
590 IF AV=155 OR SP=MAX THEN RETURN
600 AV2=AV:IF AV>64 AND AV<91 THEN AV=
AV-64
605 IF AV=46 THEN AV=30
610 IF AV=28 THEN AV=39
620 SP=SP+1:ANS$(SP,SP)=CHR$(AV):P$=CH
R$(AV):GOSUB 740
630 IF GM>0 THEN GOSUB 790
640 ANS2$(SP,SP)=CHR$(AV2)
650 NEXT R2
660 IF SP<0 THEN SP=0:NEXT R1
670 ANS$(SP,SP)=CHR$(32):P$=CHR$(32):G
OSUB 740:ANS2$(SP,SP)=CHR$(32):SP=SP-1
:
NEXT R2
730 REM ***** PRINT KEY *****
740 X=SP+XX
750 POSITION X,Y
760 IF GM>0 THEN ? #6:P$:POKE 764,255:
RETURN
770 ? P$:RETURN
780 REM **** SHOW ASC VALUE AND CHARAC
TER OF KEY SELECTED*****
790 ? "AV=";AV: ? " ";
800 ? "AV2=";AV2: ? " ";CHR$(AV2)
810 ANS2$(SP,SP)=CHR$(AV2):RETURN
820 DATA 479,55,487,95,495,111,503,247
830 POKE 756,SL/256+2
900 GRAPHICS 17:POKE 756,SL/256+2
905 ? #6: " %": ? " ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
"
910 POSITION 1,4
920 ? #6: " \000 \001 \002 \003 \004 \005
\006 \007 \008 \009 \010 \011 \012 \013
\014 \015 \016 \017 \018 \019 \020 \021
\022 \023 \024 \025 \026 \027 \028 \029
\030 \031 \032 \033 \034 \035 \036 \037
\038 \039 \040 \041 \042 \043 \044 \045
\046 \047 \048 \049 \050 \051 \052 \053
\054 \055 \056 \057 \058 \059 \060 \061
\062 \063 \064 \065 \066 \067 \068 \069
\070 \071 \072 \073 \074 \075 \076 \077
\078 \079 \080 \081 \082 \083 \084 \085
\086 \087 \088 \089 \090 \091 \092 \093
\094 \095 \096 \097 \098 \099 \100 \101
\102 \103 \104 \105 \106 \107 \108 \109
\110 \111 \112 \113 \114 \115 \116 \117
\118 \119 \120 \121 \122 \123 \124 \125
\126 \127 \128 \129 \130 \131 \132 \133
\134 \135 \136 \137 \138 \139 \140 \141
\142 \143 \144 \145 \146 \147 \148 \149
\150 \151 \152 \153 \154 \155 \156 \157
\158 \159 \160 \161 \162 \163 \164 \165
\166 \167 \168 \169 \170 \171 \172 \173
\174 \175 \176 \177 \178 \179 \180 \181
\182 \183 \184 \185 \186 \187 \188 \189
\190 \191 \192 \193 \194 \195 \196 \197
\198 \199 \200 \201 \202 \203 \204 \205
\206 \207 \208 \209 \210 \211 \212 \213
\214 \215 \216 \217 \218 \219 \220 \221
\222 \223 \224 \225 \226 \227 \228 \229
\230 \231 \232 \233 \234 \235 \236 \237
\238 \239 \240 \241 \242 \243 \244 \245
\246 \247 \248 \249 \250 \251 \252 \253
\254 \255 \256 \257 \258 \259 \260 \261
\262 \263 \264 \265 \266 \267 \268 \269
\270 \271 \272 \273 \274 \275 \276 \277
\278 \279 \280 \281 \282 \283 \284 \285
\286 \287 \288 \289 \290 \291 \292 \293
\294 \295 \296 \297 \298 \299 \300 \301
\302 \303 \304 \305 \306 \307 \308 \309
\310 \311 \312 \313 \314 \315 \316 \317
\318 \319 \320 \321 \322 \323 \324 \325
\326 \327 \328 \329 \330 \331 \332 \333
\334 \335 \336 \337 \338 \339 \340 \341
\342 \343 \344 \345 \346 \347 \348 \349
\350 \351 \352 \353 \354 \355 \356 \357
\358 \359 \360 \361 \362 \363 \364 \365
\366 \367 \368 \369 \370 \371 \372 \373
\374 \375 \376 \377 \378 \379 \380 \381
\382 \383 \384 \385 \386 \387 \388 \389
\390 \391 \392 \393 \394 \395 \396 \397
\398 \399 \400 \401 \402 \403 \404 \405
\406 \407 \408 \409 \410 \411 \412 \413
\414 \415 \416 \417 \418 \419 \420 \421
\422 \423 \424 \425 \426 \427 \428 \429
\430 \431 \432 \433 \434 \435 \436 \437
\438 \439 \440 \441 \442 \443 \444 \445
\446 \447 \448 \449 \450 \451 \452 \453
\454 \455 \456 \457 \458 \459 \460 \461
\462 \463 \464 \465 \466 \467 \468 \469
\470 \471 \472 \473 \474 \475 \476 \477
\478 \479 \480 \481 \482 \483 \484 \485
\486 \487 \488 \489 \490 \491 \492 \493
\494 \495 \496 \497 \498 \499 \500 \501
\502 \503 \504 \505 \506 \507 \508 \509
\510 \511 \512 \513 \514 \515 \516 \517
\518 \519 \520 \521 \522 \523 \524 \525
\526 \527 \528 \529 \530 \531 \532 \533
\534 \535 \536 \537 \538 \539 \540 \541
\542 \543 \544 \545 \546 \547 \548 \549
\550 \551 \552 \553 \554 \555 \556 \557
\558 \559 \560 \561 \562 \563 \564 \565
\566 \567 \568 \569 \570 \571 \572 \573
\574 \575 \576 \577 \578 \579 \580 \581
\582 \583 \584 \585 \586 \587 \588 \589
\590 \591 \592 \593 \594 \595 \596 \597
\598 \599 \600 \601 \602 \603 \604 \605
\606 \607 \608 \609 \610 \611 \612 \613
\614 \615 \616 \617 \618 \619 \620 \621
\622 \623 \624 \625 \626 \627 \628 \629
\630 \631 \632 \633 \634 \635 \636 \637
\638 \639 \640 \641 \642 \643 \644 \645
\646 \647 \648 \649 \650 \651 \652 \653
\654 \655 \656 \657 \658 \659 \660 \661
\662 \663 \664 \665 \666 \667 \668 \669
\670 \671 \672 \673 \674 \675 \676 \677
\678 \679 \680 \681 \682 \683 \684 \685
\686 \687 \688 \689 \690 \691 \692 \693
\694 \695 \696 \697 \698 \699 \700 \701
\702 \703 \704 \705 \706 \707 \708 \709
\710 \711 \712 \713 \714 \715 \716 \717
\718 \719 \720 \721 \722 \723 \724 \725
\726 \727 \728 \729 \730 \731 \732 \733
\734 \735 \736 \737 \738 \739 \740 \741
\742 \743 \744 \745 \746 \747 \748 \749
\750 \751 \752 \753 \754 \755 \756 \757
\758 \759 \760 \761 \762 \763 \764 \765
\766 \767 \768 \769 \770 \771 \772 \773
\774 \775 \776 \777 \778 \779 \780 \781
\782 \783 \784 \785 \786 \787 \788 \789
\790 \791 \792 \793 \794 \795 \796 \797
\798 \799 \800 \801 \802 \803 \804 \805
\806 \807 \808 \809 \810 \811 \812 \813
\814 \815 \816 \817 \818 \819 \820 \821
\822 \823 \824 \825 \826 \827 \828 \829
\830 \831 \832 \833 \834 \835 \836 \837
\838 \839 \840 \841 \842 \843 \844 \845
\846 \847 \848 \849 \850 \851 \852 \853
\854 \855 \856 \857 \858 \859 \860 \861
\862 \863 \864 \865 \866 \867 \868 \869
\870 \871 \872 \873 \874 \875 \876 \877
\878 \879 \880 \881 \882 \883 \884 \885
\886 \887 \888 \889 \890 \891 \892 \893
\894 \895 \896 \897 \898 \899 \900 \901
\902 \903 \904 \905 \906 \907 \908 \909
\910 \911 \912 \913 \914 \915 \916 \917
\918 \919 \920 \921 \922 \923 \924 \925
\926 \927 \928 \929 \930 \931 \932 \933
\934 \935 \936 \937 \938 \939 \940 \941
\942 \943 \944 \945 \946 \947 \948 \949
\950 \951 \952 \953 \954 \955 \956 \957
\958 \959 \960 \961 \962 \963 \964 \965
\966 \967 \968 \969 \970 \971 \972 \973
\974 \975 \976 \977 \978 \979 \980 \981
\982 \983 \984 \985 \986 \987 \988 \989
\990 \991 \992 \993 \994 \995 \996 \997
\998 \999
995 POSITION 0,23: ? #6: " ^ ^ ^ ^ ^ ^
^ ^ ^ ^
1000 FOR W=1 TO 2000:NEXT W:RETURN

```


Bob Luce: QUICKDOS

in memory Micro-DOS

```

1 REM *****
2 REM * ATARI COMPUTER ENTHUSIASTS **
3 REM * 3662 VINE MAPLE DR **
4 REM * EUGENE, OR 97405 **
5 REM * $10 YEAR **
6 REM *****
7 REM *****
8 REM *REPRINTED FROM FACE (FRESNO ATA
9 REM *RI COMPUTER SECTOR, by Bob Luce
100 REM *****
110 REM * QUICKDOS *
120 REM *****
130 OPEN #1,8,0,"D:AUTORUN.SYS"
140 TRAP 180
150 READ A
160 PUT #1,A
170 GOTO 140
180 CLOSE #1
190 ? "ALL DONE"
200 END
210 DATA 255,255,00,04,120,04,118
220 DATA 6,77,160,214,6,219,6
230 DATA 224,6,229,6,0,0,125,81
240 DATA 85,73,75,68,79,83,32,49
250 DATA 46,48,155,155,49,45,68,73
260 DATA 82,69,67,84,79,82,89,155
270 DATA 50,45,66,65,83,73,67,155
280 DATA 51,45,76,79,67,75,155,52
290 DATA 45,85,78,76,79,67,75,155
300 DATA 53,45,68,69,76,69,84,69
310 DATA 155,54,45,82,69,78,65,77
320 DATA 69,155,55,45,68,79,83,155
330 DATA 155,93,165,10,141,12,4,165
340 DATA 11,141,13,4,169,65,133,10
350 DATA 141,70,21,169,6,133,11,141
360 DATA 74,21,96,68,49,58,42,46
370 DATA 42,0,6,252,6,169,5,157
380 DATA 66,3,76,13,6,169,9,157
390 DATA 66,3,169,64,157,72,3
400 DATA 169,192,160,3,157,68,3
410 DATA 152,157,69,3,32,86,228
420 DATA 96,169,12,157,66,3,32,86
430 DATA 228,96,169,11,141,66,3,169
440 DATA 1,141,72,3,169,89,160,4
450 DATA 162,0,142,73,3,32,22,6
460 DATA 96,169,11,141,66,3,169,75
470 DATA 141,72,3,169,14,160,4,162
480 DATA 0,32,22,6,32,42,6,32
490 DATA 0,6,173,192,3,56,233,49
500 DATA 201,7,176,221,10,170,189,0
510 DATA 4,141,116,6,232,189,0,4
520 DATA 141,117,6,76,0,0,32,42
530 DATA 6,32,0,6,173,192,3,48
540 DATA 3,141,116,4,162,16,32,33
550 DATA 6,169,3,157,66,3,169,6
560 DATA 157,74,3,169,115,160,4,32

```

```

570 DATA 22,6,152,48,40,162,16,169
580 DATA 5,157,66,3,32,13,6,192
590 DATA 136,208,18,169,49,141,116,4
600 DATA 162,16,32,33,6,162,0,32
610 DATA 0,6,76,65,6,162,0,32
620 DATA 0,6,76,157,6,132,185,169
630 DATA 49,141,116,4,162,16,32,33
640 DATA 6,169,155,76,64,185,169,35
650 DATA 76,231,6,169,36,76,231,6
660 DATA 169,33,76,231,6,169,32,162
670 DATA 16,157,66,3,32,42,6,32,0,6
680 DATA 162,16,32,13,6,152,48,203
690 DATA 76,65,6,226,2,227,2,90,4

```

Assembly-Language Listing

In SynAssembler Format

```

00010 *****
00020 * MENU/COMMAND LIST *
00030 *****
00040 .OR $400
00050 .IF "D:QMENU.OBJ"
00060 *
00070 * Uses the Safe $80 Bytes
00080 * of the cassette buffer
00090 * for data and initialization
00100 * code.
00110 *
00120 COMMANDS .DA DIR ;Addresses of
00130 .DA BASIC ;Quikdos Commands
00140 .DA LOCK
00150 .DA UNLOCK
00160 .DA DELETE
00170 .DA RENAME
00180 G0DOS .HS 0000 ;to be modified
00190 MENU .HS 7D ;Clearscreen
00200 .AS "QUIKDOS 1.0"
00210 .HS 9B9B
00220 .AS "1-DIRECTORY"
00230 .HS 9B
00240 .AS "2-BASIC"
00250 .HS 9B
00260 .AS "3-LOCK"
00270 .HS 9B
00280 .AS "4-UNLOCK"
00290 .HS 9B
00300 .AS "5-DELETE"
00310 .HS 9B
00320 .AS "6-RENAME"
00330 .HS 9B

```



```

00340 .AS "7-DOS"
00350 .HS 9898
00360 MENULEN .EQ *-MENU
00370 PCHR .AS "J" ;Prompt Character
00380 PLEN .EQ *-PCHR
00390 INIT LDA DOSVEC ;Store the real DOS
00400 STA G0DOS ;vector in the
00410 LDA DOSVEC+1 ;Command Address Table
00420 STA G0DOS+1
00430 LDA #START ;Put Quikdos
00440 STA DOSVEC ;in it's place
00450 STA $1546 ;and in DOS init code
00460 LDA /START
00470 STA DOSVEC+1
00480 STA $154A
00490 RTS ;Back to DOS load
00500 DIRNAM .AS "D1:*,*" ;a few spare bytes
00510 DIRLEN .EQ *-DIRNAM ;so we'll put this here

```

```

00010 *****

```

```

00020 * *

```

```

00030 * QUIKDOS *

```

```

00040 * Written by Bob Luce *

```

```

00050 * (08/83) *

```

```

00060 * *

```

```

00070 *****

```

```

00080 *

```

```

00090 * Zero Page Equates

```

```

00100 *

```

```

00110 DOSVEC .EQ $000A

```

```

00120 DOSINI .EQ $000C

```

```

00130 *

```

```

00140 * O.S. Equates

```

```

00150 *

```

```

00160 ICCOM .EQ $0342

```

```

00170 ICSTA .EQ $0343

```

```

00180 ICBAL .EQ $0344

```

```

00190 ICBALH .EQ $0345

```

```

00200 ICBLL .EQ $0348

```

```

00210 ICBLLH .EQ $0349

```

```

00220 ICAX1 .EQ $034A

```

```

00230 ICAX2 .EQ $034B

```

```

00240 CIOV .EQ $E456

```

```

00250 BASIC .EQ $A04D

```

```

00260 INBUFF .EQ $03C0

```

```

00270 BASICERR .EQ $B940

```

```

00280 *

```

```

00290 * Misc. Equates

```

```

00300 *

```

```

00310 READ .EQ $04

```

```

00320 WRITE .EQ $08

```

```

00330 APPEND .EQ $09

```

```

00340 UPDATE .EQ $0C

```

```

00350 OPEN .EQ $03

```

```

00360 GETREC .EQ $05

```

```

00370 OPDIR .EQ $06

```

```

00380 GETCHR .EQ $07

```

```

00390 PUTREC .EQ $09

```

```

00400 PUTCHR .EQ $0B

```

```

00410 CLOSE .EQ $0C

```

```

00420 *

```

```

00430 .IN "D:QMENU.SRC"

```

```

00440 *

```

```

00450 .OR $0600

```

```

00460 .TF "D:QUIKDOS.OBJ"

```

```

00470 *

```

```

00480 *

```

```

00490 * Subroutines

```

```

00500 *

```

```

00510 * CIO Routines

```

```

00520 *

```

```

00530 * X=IOCB to use

```

```

00540 *

```

```

00550 EGETREC LDA #GETREC ;Get records

```

```

00560 STA ICCOM,X ;from D: or E:

```

```

00570 JMP CONTINUE

```

```

00580 EPUTREC LDA #PUTREC ;Put a record

```

```

00590 STA ICCOM,X ;to E:

```

```

00600 CONTINUE LDA #$40 ;Enter here to

```

```

00610 STA ICBLL,X ;use printer buffer

```

```

00620 LDA #INBUFF

```

```

00630 LDY /INBUFF

```

```

00640 CIOG0 STA ICBAL,X ;Enter with lo buffer

```

```

00650 TYA ;address in A

```

```

00660 STA ICBALH,X ;hi address in Y

```

```

00670 CIO JSR CIOV

```

```

00680 RTS

```

```

00690 *

```

```

00700 CLOSEIT LDA #CLOSE ;Close IOCB X

```

```

00710 STA ICCOM,X

```

```

00720 JSR CIOV

```

```

00730 RTS

```

```

00740 PROMPT LDA #PUTCHR ;Print prompt

```

```

00750 STA ICCOM

```

```

00760 LDA #PLEN

```

```

00770 STA ICBLL

```

```

00780 LDA #PCHR

```

```

00790 LDY /PCHR

```

```

00800 LDX #$00

```

```

00810 STX ICBLLH

```

```

00820 JSR CIOG0

```

```

00830 RTS

```

```

00840 *

```

```

00850 * Main Program Code

```

```

00860 *

```

```

00870 START LDA #PUTCHR ;Print Menu

```

```

00880 STA ICCOM

```

```

00890 LDA #MENULEN

```

```

00900 STA ICBLL

```

```

00910 LDA #MENU

```

```

00920 LDY /MENU

```

```

00930 LDX #$00

```

```

00940 JSR CIOG0

```

```

00950 JSR PROMPT ;Print Prompt

```

```

00960 JSR EGETREC ;Get user's input

```

```

00970 LDA INBUFF ;Ignore everything

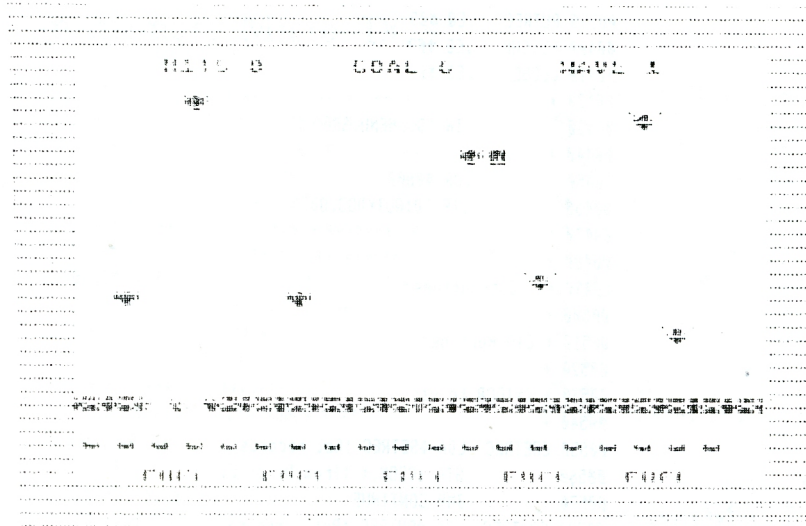
```

```

00980 SEC ;but first character

```


Stan Ockers: Missile Defense



A game that demonstrates the PM Helper of the last issue

```

1 REM *****
2 REM * ATARI COMPUTER ENTHUSIASTS **
3 REM * 3662 VINE MAPLE DR **
4 REM * EUGENE, OR 97405 **
5 REM * $10 YEAR **
6 REM *****
7 REM *****
8 REM ** MISSILE DEFENCE **
9 REM ** BY **
10 REM * STAN OCKERS **
11 REM * Oct 1983 **
12 REM *****
100 GOTO 400
200 IF MF>C1 THEN GOTO C218
202 S=PEEK(STICK0):X2=PEEK(XP02):IF S=
C7 AND X2<C192 THEN POKE XP02,X2+C8:X=
X
+C2:MCOL=MCOL+C1
204 IF S=C11 AND X2>C48 THEN POKE XP02
,X2-C8:X=X-C2:MCOL=MCOL-C1
206 IF S=C13 AND SHLD$(X+C1,X+C1)="a"
THEN SHLD$(X+C1,X+C2)=" ":POSITION C0
C19:? SHLD$;
208 IF S=C14 AND SHLD$(X+C1,X+C1)=" "
THEN SHLD$(X+C1,X+C2)="ab":POSITION C0
C19:? SHLD$;
210 IF PEEK(STR160)=C1 OR MISL$(X+C1,X
+C1)<>"c" THEN GOTO C218
216 MF=C2:MISL$(X+C1,X+C2)=" ":POSITI
ON C0,C21:? MISL$;POKE IM62,C100:POKE
YP02,C200:POKE AUDC3,140
218 IF MF<>C2 THEN GOTO C232

```

```

244 IF AMF=C0 THEN AR=INT(RND(C0)*C19)
:IF ST(AR)=C1 THEN AMF=C1:W2=C14:POKE
A
UDC1,45:POSITION X(AR),Y(AR):? AL$;
246 IF AMF=C1 THEN POKE AUDF1,W2:W2=W2
-C1:IF W2<C11 THEN POKE AUDC1,C0:AMF=C
0
:ST(AR)=C2
248 IF LAF<>C1 THEN 256
250 LAC=INT(RND(C0)*C19):LAX=LAC+LAC:I
F ST(LAC)<>2 THEN 256
254 LAF=C2:POKE IM60,BM:POKE XP00,LX(L
AC):POKE YP00,LY(LAC):ST(LAC)=C3
256 IF LAF<>C2 THEN 270
258 Y0=PEEK(YP00):IF Y0>170 AND SHLD$(
LAX+1,LAX+1)="a" THEN 266
260 IF Y0>C190 AND MISL$(LAX+C1,LAX+C1
)="c" THEN 268
262 IF Y0>C200 THEN LAF=C3:GOTO 800
264 POKE YP00,Y0+LDEL:GOTO C270
266 SHLD$(LAX+C1,LAX+C2)="rs":POSITION
C0,C19:? SHLD$;LAF=C3:POKE IM60,C20:
6
OTO C270
268 MISL$(LAX+C1,LAX+C2)=" ":POSITION
C0,C21:? MISL$;LAF=C3:POKE IM60,C20
270 IF LAF<>C3 THEN 282
272 POKE WID0,C1:I=PEEK(IM60):IF I=C20
THEN POKE IM60,C40:POKE AUDC1,140:POK
E
AUDF1,70
274 IF I=C40 THEN POKE IM60,C60:POKE A
UDF1,60
276 IF I=C60 THEN POKE XP00,C0:Y0=C0:P
OKE WID0,C0:LAF=C1:POKE AUDC1,C0
282 IF HITS=GOAL AND MF=C1 THEN 900
284 R=INT(RND(C0)*C19):IF ST(R)=3 THEN
POSITION X(R),Y(R):? " ":ST(R)=0
290 P=PEEK(RANDOM):IF P<DLY THEN R=INT
(RND(C0)*C19):R=R+R:MISL$(R+C1,R+C2)="
c
d":POSITION C0,C21:? MISL$;
294 IF P<BSP THEN BUG=BUG+C1:POKE YP01
,BUG:IF BUG>210 THEN 800
296 IF I1=C0 THEN I1=C1:POKE IM61,C200
:GOTO 300
298 IF I1=C1 THEN I1=C0:POKE IM61,180
300 GOTO C200
310 REM *****
320 REM ** MISSILE DEFENCE **
330 REM ** S.O. 9/83 **
340 REM *****
400 C0=0:C1=1:C2=2:C3=3:C7=7:C8=8:C11=
11:C13=13:C14=14:C19=19:C21=21:C48=48:
C
192=192:C214=214:C216=216
402 C218=218:C232=232:C20=20:C40=40:C6
0=60:C80=80:C100=100:C120=120:C170=170
:
C190=190:C200=200:C270=270

```


by Shane Rolin

UTILITIES

```
10340 DATA 104,160,98,162,228,169,7,76
,92,228
10342 GRAPHICS 0:POKE 752,1:?"
10350 RESTORE 10360:FOR J=1664 TO 1701
:READ A:POKE J,A:NEXT J
10360 DATA 0,0,0,0,165,165,171,161,160
,20,120,60,160,20,120,60
10365 DATA 0,0,0,0,64,152,129,0,7,20,20
,20,20,152,0,76,62,58,74,0,1,0,1
10370 POKE 1692,PEEK(1061):XPO0=1668:X
P01=1669:XPO2=1670:XPO3=1671
10380 YPO0=1684:YPO1=1685:YPO2=1686:Y
P03=1687:IMG0=1676:IMG1=1677:IMG2=1678:
1
M03=1679

10386 FOR J=0 TO 3:POKE 704+J,PEEK(169
4+J):NEXT J
10388 FOR J=0 TO 3:POKE 53256+J,PEEK(1
698+J):NEXT J
10400 PMPG=PEEK(1692):RESTORE 10420:F0
R J=0 TO 239:READ A:POKE PMPG*256+256+
J
,A:NEXT J
10410 POKE 623,1:A=USR(ADR(INVBI)):RE
TURN
10420 DATA 2,2,1,1,2,2,1,1,2,2,1,1,2,2
,1,1,2,2,1,1
10430 DATA 0,0,0,0,0,0,0,0,0,0,8,16,20,4
,0,0,0,0,0,0,0
10440 DATA 0,0,0,0,0,0,16,20,4,40,42,2
,16,16,0,0,0,0,0,0
10450 DATA 16,16,0,4,60,65,33,32,0,0,1
30,130,16,84,60,0,0,40,41,1
10460 DATA 5,2,0,5,2,0,5,2,0,5,2,0,5,2
,0,0,0,0,0,0
10470 DATA 24,24,24,90,90,126,126,90,0
,0,0,0,0,0,0,0,0,0,0,0
10480 DATA 0,0,0,0,0,0,0,0,24,24,24,24
,24,24,24,0,0,0,0
10490 DATA 0,2,1,2,7,7,2,0,0,0,0,0,0,0
,0,0,0,0,0,0
10500 DATA 17,17,27,31,21,14,14,14,14,
14,4,0,0,0,0,0,0,0,0,0
10510 DATA 40,56,124,254,84,170,0,0,0,
0,0,0,0,0,0,0,0,0,0
10520 DATA 40,56,124,254,40,84,0,0,0,0
,0,0,0,0,0,0,0,0,0,0
10530 DATA 231,195,129,0,0,36,36,36,36
,36,36,36,36,36,0,0,129,195,231
```

```
1 REM *****
2 REM * ATARI COMPUTER ENTHUSIASTS **
3 REM * 3662 VINE MAPLE DR **
4 REM * EUGENE, OR 97405 **
5 REM * $10 YEAR **
6 REM *****
7 REM *****
100 REM
110 REM ROUTINE TO LOAD IN A PICTURE
120 REM FROM THE USER'S PROGRAM.
130 REM BY SHANE ROLIN(C-SOFTWARE)
140 REM
150 REM SET FILENAME$ TO EQUAL THE
160 REM NAME OF THE FILE TO BE LOADED
170 REM IN.
180 REM
200 DIM MACHINE$(31),FILENAME$(20)
210 TRAP 500
220 GOSUB 370
230 OPEN #2,4,0,"K"
240 A=PEEK(708):B=PEEK(709):C=PEEK(710
):D=PEEK(712):MODE=7:SCRNUM=7680:E=8:X
Y=13
260 GOSUB 360
280 OPEN #1,4,0,FILENAME$
290 GET #1,F
300 MODE=8:GOSUB 350:POKE SCRADDR,F:SC
RADDR=SCRADDR+1:SCRNUM=SCRNUM-1:GOTO 3
10
310 POKE 850,7:PICTURELOAD=USR(ADR(MAC
HINE$),16,SCRADDR,SCRNUM)
320 CLOSE #1
330 REM
331 REM THIS IS WHERE THE ROUTINE TO
332 REM LOAD AND DISPLAY THE PICTURE
333 REM ENDS. HERE SHOULD BE PLACED
334 REM A DELAY STATEMENT TO HOLD THE
335 REM PICTURE ON THE SCREEN FOR AN
336 REM AMOUNT OF TIME.
350 GRAPHICS MODE+16:C=0:SCRADDR=PEEK(
88)+PEEK(89)*256
360 POKE 708,A:POKE 709,B:POKE 710,C:P
OKE 712,D:RETURN
370 FOR READR=1 TO 31
380 READ A
390 MACHINE$(READR,READR)=CHR$(A)
400 NEXT READR
410 RETURN
420 DATA 104,104,104,170,104,157
430 DATA 69,3,104,157,68,3,104
440 DATA 157,73,3,104,157,72,3
450 DATA 32,86,228,192,0,40,1,96
460 DATA 132,203,96

100 REM -----
105 REM
110 REM --DECIMAL TO HEXADECIMAL---
115 REM -----CONVERTER-----
120 REM ---ORIGINAL VERSION BY---
125 REM -----RANDY RHODES-----
130 REM --MODIFIED BY SHANE ROLIN--
135 REM -----OF C-SOFTWARE-----
140 REM
150 ? CHR$(125)
155 DIM B$(60)
160 FOR T=1 TO 60
165 READ A
170 B$(T,T)=CHR$(A)
175 NEXT T
180 DATA 104,104,141,213,0,104,141
185 DATA 212,0,160,4,173,212,0,41
190 DATA 15,78,213,0,110,212,0,78
195 DATA 213,0,110,212,0,78,213,0
200 DATA 110,212,0,78,213,0,110,212
205 DATA 0,24,105,40,201,58,40,2
210 DATA 105,6,201,71,16,6,153,202
215 DATA 0,136,208,208,96
220 ? "INPUT DECIMAL NUMBER"
225 INPUT A
230 X=USR(ADR(B$),A)
235 FOR I=203 TO 206
240 FOR T=1 TO 60:NEXT T
245 ? CHR$(PEEK(I));
250 NEXT I
255 ? :? :GOTO 220

100 REM [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
105 REM
110 REM --HEXADECIMAL TO DECIMAL--
115 REM -----CONVERTER-----
120 REM ---ORIGINAL VERSION BY---
125 REM -----RANDY RHODES-----
130 REM --MODIFIED BY SHANE ROLIN--
135 REM -----OF C-SOFTWARE-----
140 REM
145 REM [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
150 ? CHR$(125)
155 DIM B$(73),A$(10)
160 FOR T=1 TO 73
165 READ A
170 B$(T,T)=CHR$(A)
175 NEXT T
180 DATA 104,104,133,205,104,133,204
185 DATA 104,104,133,206,160,0,132
190 DATA 212,132,213,177,204,201,48
```



```

195 DATA 48,49,201,64,48,11,201,65
200 DATA 48,41,201,71,16,37,24,105
205 DATA 9,41,15,6,212,38,213,6,212
210 DATA 38,213,6,212,38,213,6,212
215 DATA 38,213,24,101,212,133,212
220 DATA 165,213,105,0,133,213,200
225 DATA 196,206,208,201,96
230 ? :? :? :? "INPUT HEXADECIMAL NUMB
ER"
235 INPUT A$
240 X=USR(ADR(B$),ADR(A$),LEN(A$))
245 ? "HEX NUMBER--";A$;" DEC NUMBER-
-";X
250 GOTO 230

```

Assembly

```

0 ;
30 *= $600 STARTING ADDR
40 ;
50 DECHEX
60 PLA DUMP # BYTES
70 PLA GET HIGH BYTE
80 STA TEMP+1 SAVE
90 PLA LO DATA BYTE
0100 STA TEMP SAVE
0110 ;
0120 LDY #4 INIT POINTER
0130 ;
0140 NXTCHR
0150 LDA TEMP GET HEX DIGIT
0155 AND #$F KEEP LOWER NIBL
0160 ;
0170 LSR TEMP+1
0180 ROR TEMP
0190 LSR TEMP+1 DIVIDE
0200 ROR TEMP REMAINING
0210 LSR TEMP+1 NUMBER BY 16
0220 ROR TEMP
0230 LSR TEMP+1
0240 ROR TEMP
0250 ;
0260 ; A HAS NUMBER TO CONVERT
0270 ;
0280 CLC CLEAR CARRY
0290 ADC #$30 CONVERT TO NUMER
0300 CMP #$3A < ASCII 9 ?
0310 BMI NOADD NO, DONE
0320 ;
0321 ; MUST BE LETTERS A - F
0322 ;
0330 ADC #6 CONV TO ASCII
0340 NOADD
0370 ;
0380 CMP #$47 GREATER THAN "F"
0390 BPL RETURN YES, QUIT
0400 ;

```

```

0410 ; SAVE CHARACTER
0420 ;
0430 STA $CA,Y
0440 DEY NEXT LOCATION
0460 BNE NXTCHR NO, GO BACK
0470 ;
0480 RETURN
0490 RTS RETURN
0495 *= 212
0500 TEMP .WORD 0
0510 .END

```

```

0420 CMP #$47 <= ASCII "F" ?
0430 BPL RETURN NO, ERROR
0440 ;
0450 ; ALPHA CHARACTER
0460 ;
0470 CLC CLEAR CARRY
0480 ADC #9 ADD ALPHA OFFSET
0490 ;
0500 NUMER
0510 AND #$F DUMP TOP BITS
0520 ASL 212 MULTIPLY CURRENT
0530 ROL 213 RETURN VALUE BY
0540 ASL 212 BY 16
0550 ROL 213
0560 ASL 212
0570 ROL 213
0580 ASL 212
0590 ROL 213

```

```

10 ; HEX TO DECIMAL CONVERSION
20 ;
30 ; CALLED FROM BASIC
40 ; X=USR(HEXDEC,STRADR,STRLN)
50 ; HEXDEC - ENTRY POINT
60 ; STRADR - ADDRESS OF HEX
70 ; STRING ARRAY
80 ; STRLEN - ARRAY LENGTH
90 ;
0100 ;
0110 ;
0120 *= $600 ORIGIN
0130 ;
0140 HEXDEC
0150 PLA DUMP # BYTES
0160 PLA ARRAY ADR (HI)
0170 STA $CD SAVE HI BYTE
0180 PLA LO BYTE ADDRESS
0190 STA $CC SAVE
0200 ;
0210 PLA LENGTH HI BYTE
0220 PLA # CHARS TO CNVRT
0230 STA $CE SAVE # CHARS
0240 ;
0250 LDY #0 CLR INDEX
0260 STY 212 CLR RETURN ARGS
0270 STY 213
0280 ;
0290 NXTCHR
0300 LDA ($CC),Y NEXT CHAR
0310 CMP #$30 < ASCII 0
0320 BMI RETURN YEP, ERROR
0330 ;
0340 CMP #$40 <= ASCII 9 ?
0350 BMI NUMER YES, MUST BE NUM
0360 ;
0370 ; TEST FOR A - F
0380 ;
0390 CMP #$41 < ASCII "A"
0400 BMI RETURN
0410 ;

```

```

0600 ;
0610 ; ADD IN THIS NIBBLE
0620 ;
0630 CLC CLEAR CARRY
0640 ADC 212 ADD TO LSB
0650 STA 212 SAVE LO BYTE
0660 LDA 213 GET HI BYTE
0670 ADC #0 ADD IN CARRY
0680 STA 213 SAVE HI BYTE
0690 ;
0700 INY NEXT CHARACTER
0710 CPY $CE DONE YET ?
0720 BNE NXTCHR NO NEXT CHAR
0730 ;
0740 RETURN
0750 RTS RETURN
0760 .END

```


MicroDos Assembly Listing Con't

```

00990 SBC #31 ;Convert it
01000 CMP #07 ;Legal?
01010 BGE START
01020 ASL ; *2 for 2 byte addresses
01030 TAX ;Use as index
01040 LDA COMMANDS,X ;Move command's
01050 STA JUMP+1 ;address
01060 INX
01070 LDA COMMANDS,X
01080 STA JUMP+2
01090 JMP $0000 ;And go for it.
01100 *
01110 * Command Modules
01120 *
01130 DIR JSR PROMPT ;Print second prompt
01140 JSR EGETREC ;Get drive #
01150 LDA INBUFF ;Get first character
01160 BMI CLOSE1 ;if >128 default to 1
01170 STA DIRNAM+1 ;else store ascii
01180 CLOSE1 LDX #10 ;Make sure it's closed
01190 JSR CLOSEIT
01200 LDA #OPEN ;Open #1
01210 STA ICCOM,X
01220 LDA #OPDIR ;for directory read
01230 STA ICAX1,X
01240 LDA #DIRNAM ;point to Dx:*.
01250 LDY /DIRNAM
01260 JSR CIOGO ;Open it
01270 TYA ;Error?
01280 BMI BAD ;Yes
01290 OK LDX #10 ;No-
01300 LDA #GETREC ;Get Directory read
01310 STA ICCOM,X
01320 JSR CONTINUE
01330 CPY #08 ;Check for EOF
01340 BNE PRINT ;Else go print it
01350 LDA #31 ;Reset default to D1
01360 STA DIRNAM+1
01370 LDX #10 ;Close #1
01380 JSR CLOSEIT
01390 LDX #00
01400 JSR EGETREC ;Wait for return
01410 JMP START
01420 PRINT LDX #00
01430 JSR EPUTREC ;Print it on screen
01440 JMP OK ;Back for more
01450 *
01460 BAD STY #09 ;So Basic can find it
01470 LDA #31 ;Reset Default
01480 STA DIRNAM+1
01490 LDX #10 ;Close #1
01500 JSR CLOSEIT
01510 LDA #9B ;And let BASIC
01520 JMP BASICERR ;do the work
01530 LOCK LDA #35 ;Special
01540 JMP NEXT ;No open required

```

```

01550 UNLOCK LDA #36
01560 JMP NEXT
01570 DELETE LDA #33
01580 JMP NEXT
01590 RENAME LDA #32
01600 NEXT LDX #10 ;Use IOCB #1
01610 STA ICCOM,X ;Store Command
01620 JSR PROMPT ;Second Prompt
01630 JSR EGETREC ;Get input to Printer buffer
01640 LDX #10 ;Just to be sure
01650 JSR CONTINUE ;File name is in Printer buffer!
01660 TYA ;Check for error
01670 BMI BAD
01680 JMP START ;ok-back to menu

```

RUTH LIST

```

10 *SQUARE
20 GR: CLEAR;
GOTO 0,0;
TURNTO 0
30 GR: 4 (DRAW
30; TURN 90)
40 E:

```

```

10 *SQUIRL
20 GR: CLEAR;
GOTO 0,0;
TURNTO 0
30 A: #A [SIDE
LENGTH
40 A: #B [ANGLE
50 A: #C [STEP
60 *DRAW
70 GR: DRAW
#A; TURN #B
80 C: #A=#A+#C
90 J: *DRAW
100 E:

```

```

TO SQUARE
REPEAT 4 [FD
30 RT 90]
END

```

```

TO POLYSPII
:SIDE :ANGLE
:INC
FD :SIDE
RT :ANGLE
POLYSPII
:SIDE + :INC
:ANGLE :INC
END

```

```

PILOT
10 *SQUARE
20 GR: CLEAR;
GOTO 0,0;
TURNTO 0
30 GR: 4 (DRAW
30; TURN 90)
40 E:
LOGO
TO SQUARE
REPEAT 4 [FD
30 RT 90]
END

```

```

PILOT
10 *SQUIRL
20 GR: CLEAR;
GOTO 0,0;
TURNTO 0
30 A: #A [SIDE
LENGTH
40 A: #B [ANGLE
50 A: #C [STEP
60 *DRAW
70 GR: DRAW
#A; TURN #B
80 C: #A=#A+#C
90 J: *DRAW
100 E:
LOGO
TO POLYSPII
:SIDE :ANGLE
:INC
FD :SIDE
RT :ANGLE
POLYSPII
:SIDE + :INC
:ANGLE :INC
END

```


PILOT ELLSWORTH

LEARNING WITH LOGO OR PICKING PILOT (A Comparison of the PILOT and LOGO Languages)

Parents, educators, and beginning programmers who own ATARI computers now find themselves faced with the dilemma of choosing the language which will best meet their needs. While both PILOT and LOGO are considered introductory languages, each has some very unique characteristics which should be taken into consideration when an "either or" choice must be made.

Perhaps, the first difference noticed between LOGO and PILOT after booting is the difference in editing procedure. Editing in both languages in the Edit Mode is very easy. PILOT, being like BASIC, accesses the Editor through the LIST command and allows full cursor control. It has the added advantage over BASIC of having an AUTO numbering and RENUMBERING function in the Editor.

The Editor is called in LOGO by the command, EDIT "name of" procedure. There is a full range of edit commands in LOGO, but no ability to edit in the immediate mode as procedures are being developed. We find this inability causes a considerably greater amount of typing and frustration to a beginning programmer, and the first plus for PILOT.

Structurally LOGO, which is a subset of LISP, is like PASCAL; PILOT is like BASIC. Both languages encourage structured programming, although PILOT is not as tightly structured as most languages and allows beginners more freedom. PILOT requires line numbers, LOGO does not. PILOT uses modules around which programs are built, while LOGO requires them. Like PASCAL or FORTH, LOGO is completely modular, and fully expandable. In LOGO a defined procedure is treated like a built in command. This ability becomes an important plus for LOGO in more complicated or creative programming.

Each of the languages has structural abilities the other lacks. PILOT has the MATCH command which easily allows the creation of interactive instructional programs or games with a minimum amount of programming or effort. LOGO has the ability to handle LISTS which allows one to write programs dealing with symbolic manipulation, or complex data structures without having to encode them in mathematical terms or strings. LOGO also has the ability to use recursion, which is the ability to define a problem in terms of itself.

This concept, which can be quite confusing, may be explained in mathematical terms as being similar to the factorial of a number, X being equal to X (X-1) factorial. For me, a nonmathematical person, it is easier to think of recursion as being like a pudding cake. Both the cake and the pudding use themselves and parts of the other in the incomplete state to come out with the finished product. This ability of LOGO makes possible very sophisticated programming. In my opinion, understanding recursion will help one to learn more of the complicated concepts in computer sciences as well as the nature of linguistics and thought processes in general.

Screen output is superior in PILOT. The screen is automatically cleared when a program is run. Text is broken only at the end of a word, making it easily read and the product neat in appearance. PILOT also has three text modes: Gr.0, and oversized text modes 1 & 2; LOGO has only Gr. 0.

Sound in both languages is good. PILOT is slightly inferior since it allows four voices and control of duration, while LOGO allows two voices with control of both volume and duration.

Both languages allow practically unlimited lengths for variable names. Neither requires dimensioning. Both languages allow global variables (constant during a program). Only LOGO allows local variables which are used in a procedure then discarded. This permits their later reuse with different meaning in the same program. PILOT uses numerical and string variables similar to BASIC, and has a maximum of 26 numerical variables per program. LOGO considers all variables as "objects", numerical variables being viewed as words made up of digits. LOGO has the ability to manipulate variables in LISTS and to use them in recursion which PILOT does not, while PILOT has the ability to combine text literals with variables without designation as text literals making it easier for a beginner to write programs with screen or printer output.

PILOT is much easier to debug than LOGO because it contains a TRACE command which allows the programmer to view the lines being used as the program progresses. This command is extremely useful. In my opinion it does not slow execution speed enough for beginners, but when the listing is interspersed with PA: commands to make the program run slowly, it becomes indispensable to beginners. PILOT also supports Remark statements, which LOGO does not, allowing explanations to be added for beginners or children.

Both languages support Turtle Graphics. LOGO has superior Turtle Graphics since up to four turtles can be visible if desired. LOGO also has a collision register, speed control, and either full or split screen graphics display. PILOT has an invisible turtle, and only split screen display; but it does have a fill command which greatly enhances graphics displays. The PILOT Turtle is very much faster than the LOGO Turtle(s).

The accompanying chart gives a thumbnail sketch of the major differences and abilities of LOGO and PILOT. In my opinion, PILOT is the language best suited to parents, educators, and those interested in programming who do not have the time or inclination to become deeply involved with programming languages and skills. It allows easy adaptation of text materials into electronic media, and programming using matching. Most instructive materials used by parents and educators, and simple game type programming written by children will fall into one of those two categories.

The question which I am inevitably asked next is: Why LOGO? LOGO is a more sophisticated language which has the ability not only of manipulating more complicated data, but in my opinion, has the ability to develop in children a more creative approach to problem solving and a more structured, scientific if you will, way of thinking. The ability to manipulate LISTS makes linguistic-type programming available as well as the beginnings of programs that "learn."

There are numerous concepts of mathematics and physics which can be taught painlessly through the use of LOGO Turtle graphics because of the inclusion of speed control and recursion. In short, I believe the LOGO type languages are the languages of the future. They may not be of necessary or even of interest to everyone.

Unlike PILOT, LOGO will take some time to learn, and I recommend persons interested to take a class of some kind in LOGO. For those who cannot, the following are recommended: **LOGO** by Harold Ableson (the blue book); **THE TURTLE'S SOURCE BOOK** by Beaden, Martin, and Muller; and **TURTLE NEWS and LOGO NEWSLETTER** published by the Young Peoples' LOGO Association, Inc., P.O. Box 855067, Richardson, Texas 75085.

The two short modules which follow are purely for demonstration of the difference in the coding of simple graphics procedures. The modules do not include printed prompts, although they require value input for the three variables used. To use the POLYSPII LOGO procedure, one must type POLYSPII followed by 3 values. To use the PILOT module, one must type U:SQUIRL then type each value followed by return when the graphics screen appears.

Next month I will continue my regular PILOT articles, including the program translated into LOGO with notes on the necessary changes. Later I will be preparing and explaining programming which uses the unique abilities of LOGO, as well as continuing to share some of the things we are doing with PILOT in our home.

LOGO - PILOT COMPARISON CHART

	LOGO	PILOT
EDITOR	X	X
Full cursor	X	
Renum and Auto	NA	X
Immediate mode	X	
SCREEN OUTPUT		
Auto clear	X	
Does not break words	X	
Multiple text modes	X	
SOUND		
Voices	2	4
Duration control	X	X
Volume control	X	
VARIABLES		
Local variables	X	
Global variables	X	X
No dimensioning	X	X
Unlimited name length	X	X
Maximum # variables	X	
Mix with text literals	X	
Data Types		
Numeric	NA	X
Strings	NA	X
Words	X	NA
Lists	X	NA
DEBUGGING		
Trace command	X	
Remark statements	X	
STRUCTURE		
Line numbers	X	
Modular	X	Not completely
Encourage structure Best	Good	
Recursion	X	
List processing	X	
"Matching"function	X	
Data manipulating	X	
Random Access	None	None
TURTLE GRAPHICS		
Visible Turtle	X	
Multiple Turtles	X	
Speed Control	X	
Collision register	X	
Fast execution	X	
Fill command	X	
Full screen graphics	X	
Split screen graph.	X	X

ROCKERS

Missile Defense

Here's a program to demonstrate the use of PMHelper 2.0 from last month. I've restricted myself to using Basic except for the VBI routine produced by PMHelper and something to move the character set. All sorts of techniques were tried to increase the speed of the program, but there were quite a few decisions involved as you can see by reading the line-by-line breakdown of the main loop.

Your job is to protect your fuel supplies from approaching aliens. Move the joystick left and right to select a missile and press the button to fire. Be sure and remove the shield above your missile first by pulling back on the joystick. It can be replaced (if not destroyed) by pushing the stick forward. Your missiles will be replaced at random. Missiles from the aliens will destroy first the shield, then the rockets and finally the fuel which will end the game.

With each wave you will have a goal of hits to reach. Once the goal is reached, a new wave will be started with all your shields and missiles replaced.

The object is to go through as many waves as possible. Oh yes, there is an indestructible enemy 'bug' descending on you. If it reaches your fuel supply, the game is over.

A comment on using PMHelper 2.0 ... If the module is used last, memory must be lowered in 8 page increments to make room for items above it. I found out the hard way that the module clears one page above itself (a bug). In the case of this program I've just put the character set in the top 4 of the 8 pages and not bothered to fix the bug.

Lines 10000 and up are the PM module. If you have it already all you have to change is the data in lines 10360, 10365 and 10420 through 10530

Program Structure

Missile Defense

```
200 If missile active, skip
202 Ck joystick, move pointer right
204 Move pointer left
206 If down and shield in place, remove
208 If up and shield removed, replace
210 Skip if trig. not pressed or no missile
216 Else activate missile
218 Skip if missile not in flight
220 Missile moved up screen
222 If off top, put pointer back
224 If hit shield, print destroyed shield and activate explosion
226 Ck if hit active alien, if not skip
228 Remove alien, activate explosion, increment hits
230 Ck if hit alien missile, if so activate explosion of both missiles
232 If missile explosion not active, skip
234 1st part of missile explosion
236 2nd part of explosion
238 3rd part of explosion
240 End of explosion, back to pointer
242 Activate alien depending on delay and condition of alien
244 Materialize alien at random, only if activated
246 If materializing, generate sounds
248 If not available to fire, skip
250-254 Activate alien missile at random, alien must be there
256 Skip if alien missile not moving
258 Check for hit of shield
260 Check for hit of missile
262 Check for hit of fuel
264 Advance alien missile
266 Hit shield, activate explosion
268 Hit missile, activate explosion
270 Skip if explosion not active
272 1st part of alien missile explosion
274 2nd part of explosion
276 3rd part of explosion
282 Check for goal reached
284 Remove alien at random
290 Add a missile at random
294 Advance bug, check for reaching fuel
296-298 Alternate bug images
300 Back to start of loop
```

STEALING THE DOS VECTOR

(And getting away with it!)

Pointers, vectors, what the heck are they talking about? And why do I want to steal a vector? Well, I want to show you how we can use the DOS command vector at memory locations 10 and 11 to install a memory resident utility to perform many of the most used DOS commands, without having to wait for DUP.SYS to load in, and without losing whatever BASIC program we have in memory at the time.

There are many locations in the Atari's memory which we call pointers or vectors. These are simply two bytes containing an address. This address is usually the beginning of a machine language routine to be performed. Such a vector exists at memory locations 10 and 11, the DOS vector, or DOSVEC for short. This is the address the computer goes to whenever we type the command "DOS" from the keyboard.

Since Atari Basic is in ROM memory, so are most of its pointers for its different commands, i.e., LIST, RUN, etc. Only one command, the DOS command, has a vector in RAM. Since it is in RAM memory, we can change it to whatever we please.

For those of you who are still having a little trouble understanding, try to imagine the computer as a very forgetful person. So forgetful in fact, that he has to keep the addresses of where he lives, where he goes to school, what number to use when he calls home. ... OK? Well, what happens to this forgetful person when we sneak in and steal the address for where he goes to school and leave the address for a local bar in its place? School takes on quite different meaning!

This is precisely what we are doing in our example program, Quikdos. All we need to do is replace the address in DOSVEC with the beginning address of our program. That way, whenever we type DOS from the keyboard we will go directly to Quikdos. Voila, a simple way to hook extra little machine language utilities into BASIC without having to remember some obscure USR command.

There is one small problem to this approach. Whenever the RESET key is pressed, the operating system cleans up its pointers and resets them to their original values. Like I said, just a small problem. The values to use to reset DOSVEC are also in RAM memory at addresses 5446 (lo) and 5450 (hi). If we change those addresses too, even a system reset can't wipe out what we have done! What power!

You might want to try typing in the following lines of BASIC code just for fun. To illustrate the use of the DOS vector, let's detour it right back to BASIC. Not of much use, but it does illustrate a point.

```
10 REM BASIC starts at $A000 (40960)
20 REM Calculate pokes
30 ADDRESS = 40960
40 HI = INT(ADDRESS/256)
50 LO = ADDRESS-(HI*256)
60 POKE 10,LO:POKE 11,HI
70 POKE 5446,LO:POKE 5450,HI
80 END
```

After typing in these lines, (you can, of course, leave the REM statements out), run the program and then type DOS and hit the return key. If all goes well, all you will get is the READY prompt. Type LIST and you'll see the program is now gone. That is because 40960 is the BASIC coldstart address and BASIC has reset its program pointers. Oh well, you didn't want to save the program anyway! Try the system reset key now and then try DOS again. There is no escape. The real DOS is apparently gone. It isn't really gone, but until we either change DOSVEC back or power down, it can't be found.

Now for the example program Quikdos.

Quikdos is a memory resident utility which provides immediate access to the most used DOS commands without destroying the BASIC program currently in memory. I don't know about you, but most of the time when I go to DOS, all I want is a directory of the disk files. But unless I have a MEM.SAV file on the disk, I lose my BASIC program in memory. And if I do have a MEM.SAV file, it takes an unacceptable amount of time just to get a disk directory. Enter Quikdos.

Typing in Quikdos.

Format a diskette and write DOS files to it. Make sure you are using DOS 2.0!

Type in the BASIC loader program and SAVE IT TO THE DISK. This is very important since just one wrong number in the data statements could cause your computer to go to never-never land and you will have to power down, and all the work goes down the drain.

Make sure the same diskette we just prepared is in the drive and run the program. If all goes well, you should hear the disk drive start up, write, and then the words ALL DONE will be printed on the screen.

Once again, make sure the BASIC program has been saved to the disk and turn the computer off and then back on again.

You should hear all the normal things happen and the READY prompt appear on the screen. Quikdos is now installed and ready to use.

Using Quikdos.

Upon typing the DOS command from BASIC, you will be presented with the Quikdos menu and the right bracket prompt will show you Quikdos is waiting for your command.

There are seven Quikdos commands.

1. Disk Directory
2. Return to Basic
3. Lock a file
4. Unlock a file
5. Delete a file
6. Rename a file
7. Go to the Atari DOS menu.

At this point simply enter the number corresponding to the command you want. If the command is anything other than 2-BASIC or 7-DOS, you will be presented with a second right bracket prompt. For the Directory command, enter the number of the drive you wish to read or just hit RETURN for drive one. After displaying the disk directory, the display will pause until you hit RETURN again. Then the screen will clear and the Quikdos menu will reappear. If you make an error, the error number will be displayed and you will be returned to BASIC.

For the commands Lock, Unlock, Delete, and Rename, the second bracket prompt means Quikdos is waiting for you to enter the name of the file to be processed. Enter the file name exactly like the Atari DOS manual teaches for its corresponding DOS commands. Don't forget the D: preceding the name. For the rename command, the first filename should be followed by a comma followed by the new name (without the D:). Once again, any error will display the appropriate error number and return you to BASIC. Wildcard characters can be used, but be aware that Quikdos will not ask you to verify when deleting a file. This feature was left out due to memory considerations.

Should you wish to use the Atari DOS for copying files or whatever, just use command #7.

BEWARE!!

Writing DOS files while Quikdos is still connected can be disastrous. Dos will permanently change its vectors to Quikdos' and command #7 will send you right back to Quikdos. If you must write DOS files, boot up with a "clean" version of DOS.

Quikdos uses all of page 6, the cassette buffer, and the printer buffer for input. This was done to avoid using any of BASIC's program area. If any application program uses page 6 for any reason, Quikdos will be mutilated and could crash the system. Using the cassette with Quikdos will also cause instant death when the DOS command is executed. Since the printer buffer is used for input only, use of the printer will not adversely affect Quikdos. The Monkey Wrench from Eastern House Software will not function with Quikdos since it uses page six for variable storage.

There are still many things other than just executing DOS commands, though, that can be hooked into BASIC via the DOS vector. Possibly many of you have ideas for programming utilities you might find useful. Let's hear from you!

— Bob Luce

UPPERCASE AND LOWERCASE

IN GRAPHICS ONE AND TWO

In the last issue of the newsletter, I mentioned it is possible to have both uppercase and lowercase letters at the same time in graphics modes one and two. Since then several of you have asked how it can be done, so I have put together a short program to explain how it's done. In the process I got a little carried away, and I have also included a subroutine to check user input for correct responses, and a backspacing routine for graphics one and two.

I am going to go through the program explaining the major processes pretty much line by line. For some of the more advanced programmers out there this may seem a bit too detailed, but bear with the rest of us and we'll all learn together. If you have a faster or better way to do this let me know.

LINES 10-130 are the dimensioning and initializing section.

30 opens the keyboard to be read for input.

40 moves RAMTOP to reserve memory for the character set to be moved.

60 turns the cursor off.

90 moves the entire character set.

100 replaces the control graphics characters with the uppercase characters.

110 moves the punctuation marks so they may be used.

120 makes the first character (ATASCII 0) a blank. See what happens if you remove this line.

LINES 140-370 are the main program lines. Here is where your program will be.

170 sets the parameters for an input from the user. See line 300 also.

190 tells the computer where the new character set is located. This line must be repeated everytime a GRAPHIC statement is made.

200-230 here the print statements are entered just as they normally are to produce multiple colors in graphics one and two.

240-270 here the print statements are altered to use the new character set. These will produce upper and lower case letters at the same time by using control graphics characters in place of capital letters. When typing the print statements, you must use the corresponding graphics key in place of the desired capital letter or punctuation. Change the text of line 240 using this method and watch the results.

290 prints each character on the screen one at a time at location 1,5, and at the same time prints the corresponding ATASCII number in the text window. Note: when character 125 is printed it will clear the screen.

If you have the patience this line will enable you to chart where each character and color group is located by ATASCII value. But if you have access to a copy of "Your Atari Computer" it is easier to look on page 345. They have done all the work for you, and it helps take some of the mystery out of the moved characters. Also, look at Appendix D to find out what keystrokes produce which characters.

300 is a text line asking for an input from the user. It asks a question and then sets the parameters for the input. MAX = maximum length of the answer. XX = X co-ordinate of the position statement. Y = the Y co-ordinate. NUM or LTR determines whether the answer is to be uppercase only, upper and lowercase, or numbers only. NUM = 1 for numbers. LTR = 1 for uppercase. LTR = 2 for upper and lowercase. GOSUB either 380 for letters or 390 for numbers. The POKE is to clear the last key pressed.

340 prints the answer in the altered character set and the true character set. The true value of the answer is stored in ANS2\$.

350 and 360 accept the ending flag.

400 and 420 disables the break key.

430 clears the last key pressed and turns on either the uppercase or lowercase character set.

460 gets a key and looks for the back space key.

470-550 sets the parameters for which keys are to be accepted. I have set the parameters as shown for a program I am working on. You can set them to accept any key you want by stating its ATASCII number as shown.

570 alters the ATASCII value of the apostrophe for the new character set.

600 alters the ATASCII value for the capital letters.

620 counts the letters and positions them in a string.

LINES 660-720 are the back spacing routine for graphics one and two.

LINES 730-770 contain the subroutine to position and print each character on the screen in any of the text graphics modes. It can also be used to print whole statements as it was in line 360. In a long program with many position statements this can save memory.

780-710 is used to print ANS\$ and ANS2\$ and their ATASCII values in the text window.

There are some drawbacks to redefining the character set as it was done here. The main disadvantage is we've lost the ability to easily display numbers and graphics characters. If you need to display numbers and graphic characters, you will have to use the standard approach to text display when in graphic modes one and two.

In case you were wondering why anyone might want to go to all this trouble just to have upper and lowercase on the same screen, and be very selective in what keys the user is allowed to enter, I'll give you something else to think about. We, as a users group, are trying to increase the quality of the software available to us, and it's little things like use of upper and lowercase letters which make a program visually and gramatically correct. Checking each key for correct input means a program will be harder to crash, and incorrect information cannot be entered.

Educational software is the fastest growing field in home computer software, and if we let the writers know now we are not going to use poorly written programs, we may be able to steer them in the right direction from the beginning. By learning what can be done with our computers we will be less likely to accept second rate programs.

I hope you have been able to follow what was going on. If you have learned something this has really been worthwhile. For a more complete explanation on redefining character sets read chapter three of "Computer's First Book of ATARI Graphics". Questions, comments, and curses will gladly be accepted and answered as soon as possible.

—Robert Browning

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED
— Bonded for your protection —
PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services
Inc.

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

Best of ACE books

Volume 1 are bound issues of the ACE Newsletter from the first issue, Oct 80 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kaneohe, HI 96744

SortFinder 1.2
Composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only \$6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$10 a year for U.S., and \$20 a year Overseas Air-mail and include about 10 issues a year of the ACE Newsletter.
Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

****President—** Kirt Stockwell, 4325 Sean , Eugene, OR 97402
503-689-5355

Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, Or 97405 /
503-686-1490

Secretary— Charles Andrews, POB 1613, Eugene, Or 974401613
503-747-9892

Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, Or 97404 ,
(503)689-7106.

Editors— Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405
503-344-6193

Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405
503-484-9925

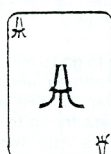
E.R.A.C.E. (Education SIG Editor)— Ali Erickson, 295 Foxtail Dr.,
Eugene, Or 97405 /503-687-1133

E.R.A.C.E. Corresponding Secretary— Robert Browning, 90 W. Myoak,
Eugene, OR 97404 (503)689-1513.

Send 27 cents in stamps or coin (50c overseas) to the Ness' for the new, updated ACE Library List —new in Oct!!

Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives with an ATR 8000 interface, 2 double density Percom disk drives, an Epson MX80 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Hubbard, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.



ATARI
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL